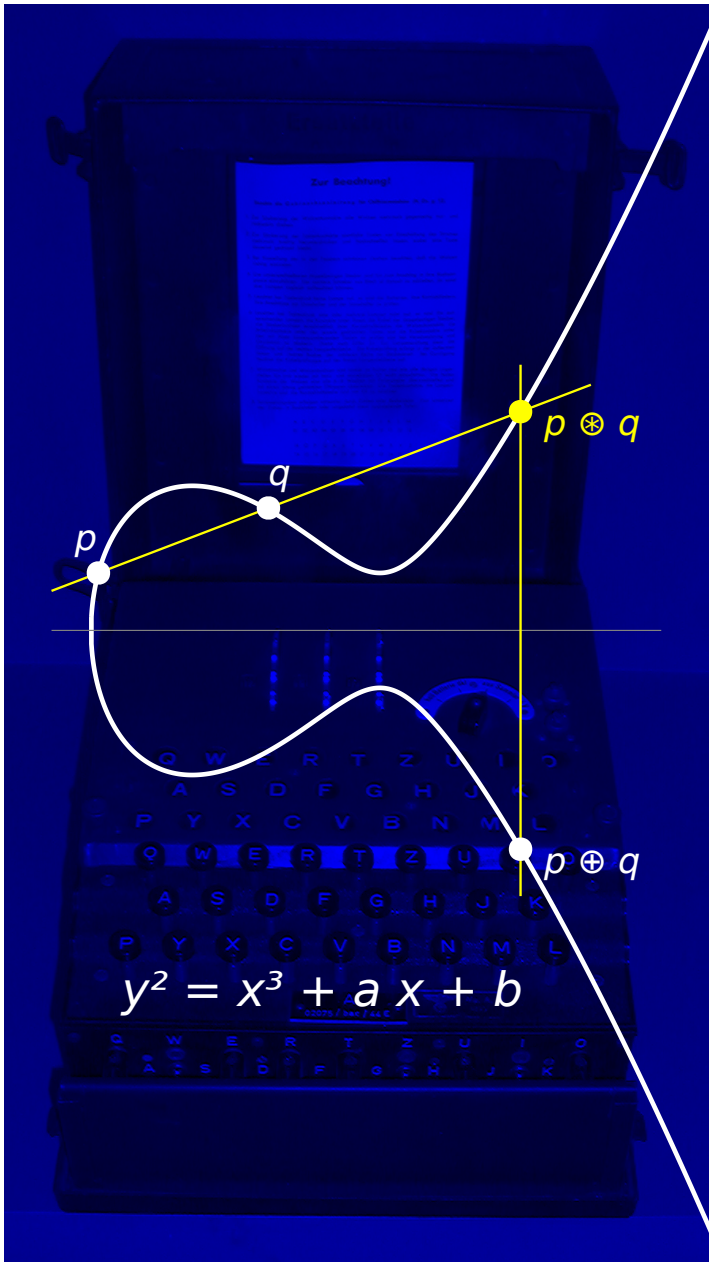


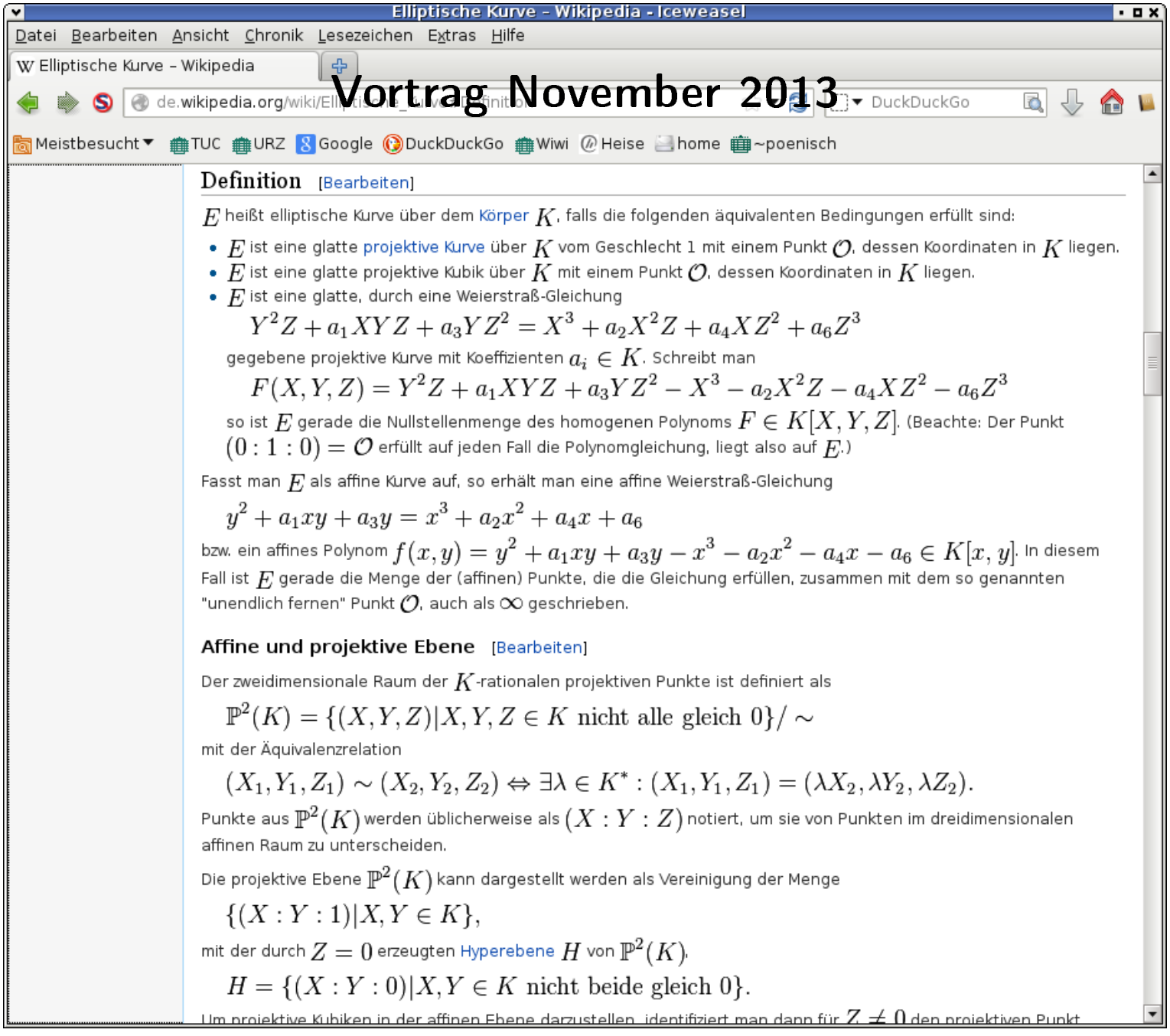
Kryptographie mit elliptischen Kurven

Versuch einer Erklärung

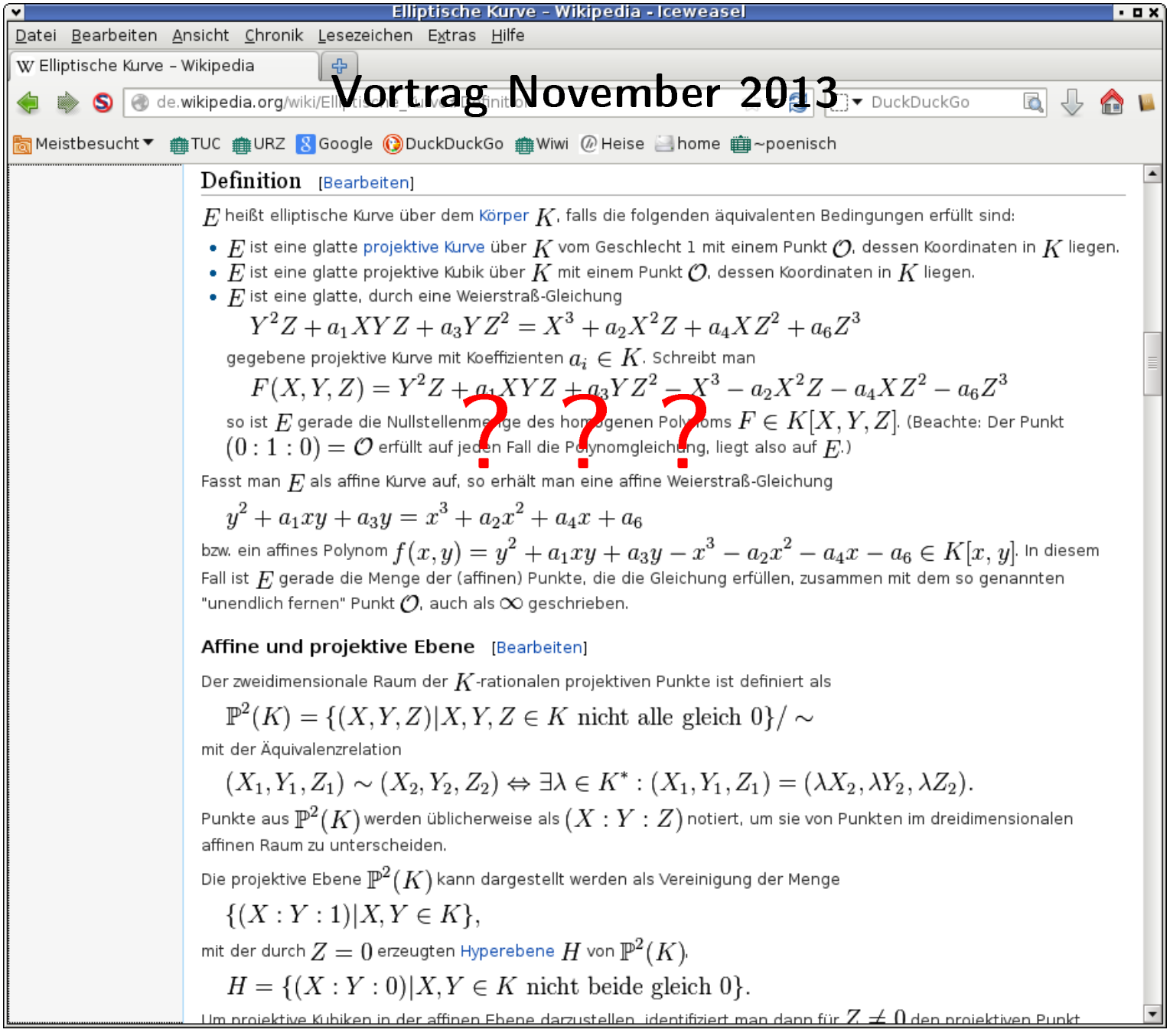
Jens Pönisch

2014-11-25

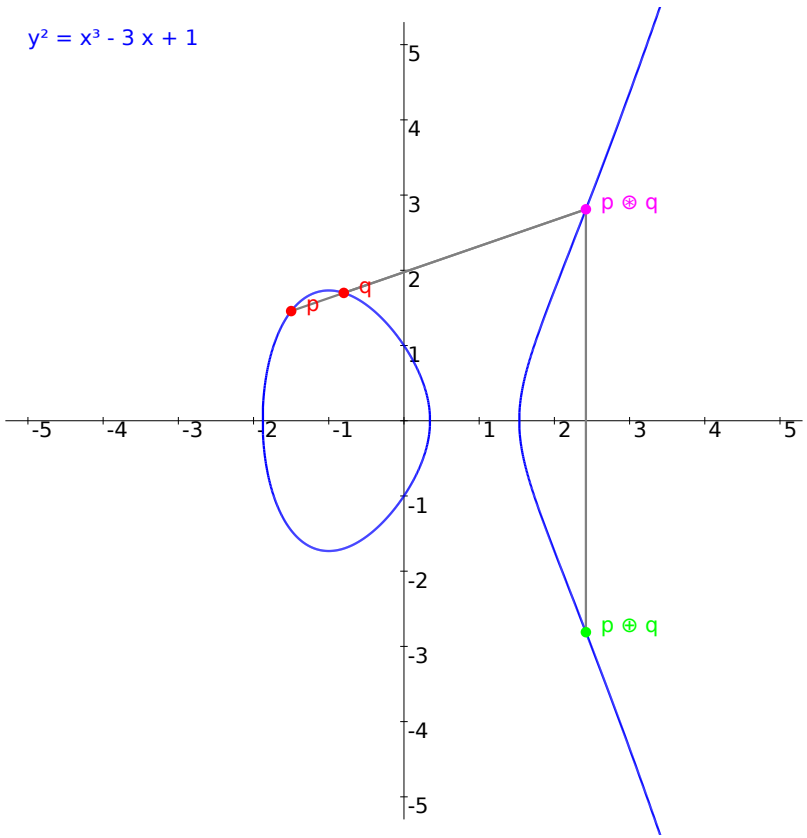
20 Fragen zur Krypographie . . .



20 Fragen zur Krypographie . . .

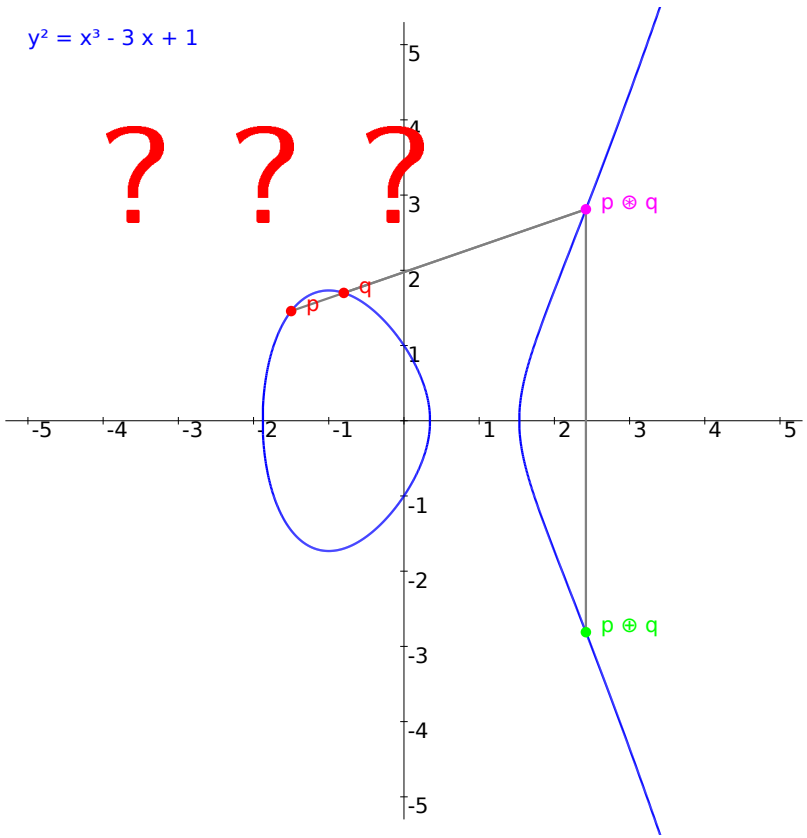


Erklärung in populärwissenschaftlichem Beitrag



- Definieren Kurve $y^2 = x^3 + a \cdot x + b$
- Definieren Punktaddition nach Bild
- Entscheidend ist Anzahl der rationalen Punkte der Kurve

Erklärung in populärwissenschaftlichem Beitrag



- Definieren Kurve $y^2 = x^3 + a \cdot x + b$
- Definieren Punktaddition nach Bild
- Entscheidend ist Anzahl der rationalen Punkte der Kurve
- Warum gibt es überhaupt rationale Punkte?
- Warum entstehen bei der Operation wieder rationale Punkte?
- Was nützt uns das?

Ziel unserer Betrachtung

- Alice und Bob wollen sich über öffentlichen Kanal *vertraulich* unterhalten.
- Eve kann die gesamte Kommunikation mitlesen.
- Alice und Bob müssen also über den öffentlichen Kanal ein Geheimnis austauschen, ohne das Eve dies erfährt.
- Alice und Bob wissen, dass sie tatsächlich mit dem richtigen Partner kommunizieren.
- $\Longrightarrow$ Diffie-Hellman-Schlüsseltausch.
- Eigentliche Verschlüsselung der Nachricht beruht auf dieser Idee.

Inhalt

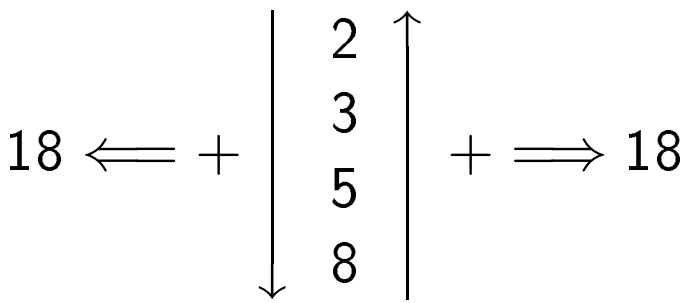
Motivation

Versuch zu verstehen, wie «Rechnen» eigentlich funktioniert.

Daraus neue Zahlen oder Rechenoperationen ableiten.

Eigenschaften Addition

- Beliebige Zahlen können addiert werden – Ergebnis ist wieder Zahl.
- Bei gleichen Summanden kommt immer das gleiche heraus.
- Wieviel muss zu a addiert werden, um b zu erhalten?
  Finden wir immer eine Lösung?
- Reihenfolge spielt keine Rolle:



Eigenschaften für neue Zahlen / Rechenoperationen schwer zu prüfen:
$\Longrightarrow$ Suche nach einfacheren Regeln!

(Abelsche) Gruppe

- Wenn zwei Zahlen zur Gruppe gehören, dann auch ihre Summe.
- Die Summe ist eindeutig.
- Es gibt eine Null: $a + 0 = 0 + a = a$
- Jede Zahl a hat eine *inverse* Zahl $-a$: $a + (-a) = 0$
- Es gilt das Assoziativgesetz: $(a + b) + c = a + (b + c)$
- (zusätzlich) Kommutativgesetz: $a + b = b + a$

Kurzschreibweise: $a + (-b) =: a - b$
Subtraktion ist nicht assoziativ und kommutativ!

- Wenn zwei Zahlen zur Gruppe gehören, dann auch ihr Produkt.
- Das Produkt ist eindeutig.
- Es gibt eine Eins: $a \cdot 1 = 1 \cdot a = a$
- Jede Zahl a hat eine *inverse* Zahl a^{-1}: $a \cdot -a^{-1} = 1$
- Es gilt das Assoziativgesetz: $(a \cdot b) \cdot c = a \cdot (b \cdot c)$
- (zusätzlich) Kommutativgesetz: $a \cdot b = b \cdot a$

Kurzschreibweise: $a \cdot b^{-1} =: a/b$
Division ist nicht assoziativ und kommutativ!

Beispiele Gruppen

Bezüglich Addition

- ~~natürliche Zahlen~~
- ~~positive rationale Zahlen~~
- ganze Zahlen
- rationale Zahlen
- reelle Zahlen
- komplexe Zahlen

Bezüglich Multiplikation

- ~~natürliche Zahlen~~
- positive rationale Zahlen
- ~~ganze Zahlen~~
- rationale Zahlen *ohne Null*
- reelle Zahlen *ohne Null*
- komplexe Zahlen *ohne Null*

Elliptische Kurven: «Zahlen» sind Punkte auf der Kurve, «Addition» ist geometrische Konstruktion.

Körper

Vereinigen von Addition und Multiplikation, um «normal» rechnen zu können.

- $0 \neq 1$.
- Die Zahlen des Körpers sind Gruppe bezüglich der Addition.
- Die Zahlen des Körpers *ohne die Null* sind Gruppe bezüglich der Multiplikation.
- Es gilt das Distributivgesetz: $a \cdot (b + c) = a \cdot b + a \cdot c$

Wir können dann «ganz normal» mit solchen Zahlen in allen vier Grundrechenarten rechnen.

Körper

Vereinigen von Addition und Multiplikation, um «normal» rechnen zu können.

- $0 \neq 1$.
- Die Zahlen des Körpers sind Gruppe bezüglich der Addition.
- Die Zahlen des Körpers *ohne die Null* sind Gruppe bezüglich der Multiplikation.
- Es gilt das Distributivgesetz: $a \cdot (b + c) = a \cdot b + a \cdot c$

Wir können dann «ganz normal» mit solchen Zahlen in allen vier Grundrechenarten rechnen.

Diese wenigen Forderungen genügen, um z. B.

- Gleichungssysteme lösen zu können (Gaußverfahren),
- mit Polynomen rechnen zu können (Nullstellen).

Körper

Vereinigen von Addition und Multiplikation, um «normal» rechnen zu können.

- $0 \neq 1$.
- Die Zahlen des Körpers sind Gruppe bezüglich der Addition.
- Die Zahlen des Körpers *ohne die Null* sind Gruppe bezüglich der Multiplikation.
- Es gilt das Distributivgesetz: $a \cdot (b + c) = a \cdot b + a \cdot c$

Wir können dann «ganz normal» mit solchen Zahlen in allen vier Grundrechenarten rechnen.

Diese wenigen Forderungen genügen, um z. B.

- Gleichungssysteme lösen zu können (Gaußverfahren),
- mit Polynomen rechnen zu können (Nullstellen).

Motivation: Parameter der Kurvengleichung, Punktkoordinaten sind Zahlen aus Körper.

Beispiele von Körpern

- ~~natürliche Zahlen~~ (Subtraktion, Division geht nicht immer)
- ~~ganze Zahlen~~ (Division geht nicht immer)
- rationale Zahlen
- reelle Zahlen
- komplexe Zahlen

Probleme

Alle bisherigen Körper enthalten unendlich viele Zahlen.

Auf dem Computer:

- Nicht alle Zahlen können dargestellt werden.
- Zahlen können nicht immer exakt dargestellt werden.
- Rechnen kann zu Ausnahmen (Überlauf) führen.
- Rundungsfehler.

Problem bei kryptographischen Rechnungen!

$\Longrightarrow$ Gibt es auch Körper mit endlich vielen Zahlen?

(Was ist eigentlich der kleinstmögliche Körper?)

Primzahlkörper

- Nehmen beliebige Primzahl p, z. B. 7.
- Unsere Zahlen sind nun $0, 1, \ldots, p-1$, z. B. $0, 1, 2, 3, 4, 5, 6$
- Addition, Subtraktion, Multiplikation zunächst ganz normal:
  Wenn Wert größer oder gleich p oder negativ, solange p addieren bzw. subtrahieren, bis gültiger Wert.
  Beispiel: $3 + 5 = 8(-7) = 1$, $2 - 6 = -4(+7) = 3$
  Beim Programmieren: Division mit Rest («%»)
- Division: Multiplikation mit inverser Zahl: $a/b = a \cdot b^{-1}$
  Kann mit Euklidschen Algorithmus leicht bestimmt werden.

Bezeichnung: $\mathbf{Z}_p$ oder $GF(p)$

Primzahlkörper

- Nehmen beliebige Primzahl p, z. B. 7.
- Unsere Zahlen sind nun $0, 1, \ldots, p-1$, z. B. $0, 1, 2, 3, 4, 5, 6$
- Addition, Subtraktion, Multiplikation zunächst ganz normal: Wenn Wert größer oder gleich p oder negativ, solange p addieren bzw. subtrahieren, bis gültiger Wert.
  Beispiel: $3 + 5 = 8(-7) = 1$, $2 - 6 = -4(+7) = 3$
  Beim Programmieren: Division mit Rest («%»)
- Division: Multiplikation mit inverser Zahl: $a/b = a \cdot b^{-1}$
  Kann mit Euklidschen Algorithmus leicht bestimmt werden.

Bezeichnung: $\mathbf{Z}_p$ oder $GF(p)$

Körper hat nur p viele Zahlen, alle Rechnungen sind exakt!

Lange bekannt, aber vor Computerkryptographie ohne jeden praktischen Nutzen!

Hauptgrund: Es gibt hier «schwierige» Probleme.

Beispiel $\mathbb{Z}_7$

Addition:

$+$	1	2	3	4	5	6
1	2	3	4	5	6	0
2	3	4	5	6	0	1
3	4	5	6	0	1	2
4	5	6	0	1	2	3
5	6	0	1	2	3	4
6	0	1	2	3	4	5

Multiplikation:

$\cdot$	1	2	3	4	5	6
1	1	2	3	4	5	6
2	2	4	6	1	3	5
3	3	6	2	5	1	4
4	4	1	5	2	6	3
5	5	3	1	6	4	2
6	6	5	4	3	2	1

Die Rechnungen mit 0 wurden weggelassen.

Beispielcode: Rechnen in $\mathbf{Z}_p$

```
class Zp(object):
  def __init__(self, v, p):
    self.v = v
    self.p = p

  def inv(self):
    r0, t0 = self.p, 0
    r1, t1 = self.v % self.p, 1
    while r1 != 1:
      f = r0 // r1
      rn = r0 - f*r1
      tn = t0 - f*t1
      r0, r1 = r1, rn
      t0, t1 = t1, tn
    return Zp(t1 % self.p, self.p)

  def __add__(self, a): return Zp((self.v+a.v)%self.p, self.p)

  def __sub__(self, a): return Zp((self.v-a.v)%self.p, self.p)

  def __mul__(self, a): return Zp((self.v*a.v)%self.p, self.p)

  def __div__(self, a): return self*a.inv()
```

Motivation

Für Kryptographie: Suche nach einer Funktion:

- die sich leicht berechnen lässt (Verschlüsselung),
- deren Umkehrfunktion äußerst schwierig oder gar nicht berechenbar ist (Kryptoanalyse, Brechen der Verschlüsselung).

Potenzen

Ausgangspunkt: Gruppe mit einer Operation $\diamond$ und natürliche Zahl n, a Zahl aus Gruppe.

Bezeichnen:

$$a(\diamond)^n = \underbrace{a \diamond a \diamond \cdots \diamond a}_{n-\text{mal}}$$

Zusätzlich legen wir fest: $a(\diamond)^0$ ist das neutrale Element.

Für unsere bisherigen Operationen $+$ und $\cdot$:

$$a(+)^n = n \cdot a$$
$$a(\cdot)^n = a^n$$

Rechenaufwand

n wird später unser geheimer Schlüssel und sehr groß (1000-stellige Zahl)

Wie schnell kann $a(\diamond)^n$ berechnet werden?

Rechenaufwand

n wird später unser geheimer Schlüssel und sehr groß (1000-stellige Zahl)

Wie schnell kann $a(\diamond)^n$ berechnet werden?

- $n - 1$ Operationen?
  $\Longrightarrow$ mehrere Urknallperioden Rechenzeit

Rechenaufwand

n wird später unser geheimer Schlüssel und sehr groß (1000-stellige Zahl)

Wie schnell kann $a(\diamond)^n$ berechnet werden?

- $n - 1$ Operationen?
  $\Longrightarrow$ mehrere Urknallperioden Rechenzeit
- *Nein*: Russische Bauernmethode (schnelles Potenzieren):
  Zahl der Operationen ist linear zur Stellenzahl $(2 \cdot \log_2 n)$

Diskreter Logarithmus

Umkehrfunktion zur Potenzfunktion

a und b aus Gruppe vorgegeben.
Für welche *natürliche* Zahl x gilt

$$a(\diamond)^x = b \quad ?$$

Schreibweise:

$$x = \log(\diamond)_a b$$

Rechenaufwand für rationale Zahlen?

Operation $+$:

$$a(+)^x = a \cdot x = b$$
$$x = b/a$$

Berechnung Logarithmus

Betrachten nun Multiplikation

- Alle $x = 1, 2, \ldots$ durchprobieren?
  $\Longrightarrow$ dauert *sehr* lange für große n

$a = 6$

n	a^n
1	6
2	36
3	216
4	1296
5	7776
6	46656

Berechnung Logarithmus

Betrachten nun Multiplikation

- Alle $x = 1, 2, \ldots$ durchprobieren?
  $\Longrightarrow$ dauert *sehr* lange für große n
- Potenzfolge ist monoton $\Longrightarrow$ Binäre Suche!

Diskreter Logarithmus kann schnell berechnet werden, kryptographisch ungeeignet!

$a = 6$

n	a^n
1	6
2	36
3	216
4	1296
5	7776
6	46656

Diskreter Logarithmus im endlichen Körper

$a = 6,\ p = 11$

n	$a(+)^n$	$a(\cdot)^n$
1	6	6
2	1	3
3	7	7
4	2	9
5	8	10
6	3	5
7	9	8
8	4	4
9	10	2
10	5	1
11	0	6

Beobachtungen:

- Zahlen sind nicht mehr sortiert
  $\Longrightarrow$ Binäre Suche nicht mehr möglich
- Es kommen alle Zahlen der Gruppe vor:
  (Muss das so sein?)

Betrachten $+$:
Bei rationalen Zahlen: $\log(+)_a b = b/a$
Probieren hier:

$$\log(+)_6 10 = 10/6 = 10 \cdot 6^{-1} = 10 \cdot 2 = 9$$

Gilt immer!
$\Longrightarrow$ Diskreter Logarithmus für $+$ auch in $\mathbf{Z}_p$ einfach.

Diskreter Logarithmus bei Multiplikation

Im endlichen Körper kein schnelles Verfahren bekannt.

Für Kryptographie geeignet?

Problem (p ist wieder 11, a ist 6, 3 bzw. 10):

n	$6(\cdot)^n$	$3(\cdot)^n$	$10(\cdot)^n$
1	6	3	10
2	3	9	1
3	7	5	10
4	9	4	1
5	10	1	10
6	5	3	1
7	8	9	10
8	4	5	1
9	2	4	10
10	1	1	1

Diskreter Logarithmus bei Multiplikation

Im endlichen Körper kein schnelles Verfahren bekannt.

Für Kryptographie geeignet?

Problem (p ist wieder 11, a ist 6, 3 bzw. 10):

n	$6(\cdot)^n$	$3(\cdot)^n$	$10(\cdot)^n$
1	6	3	10
2	3	9	1
3	7	5	10
4	9	4	1
5	10	1	10
6	5	3	1
7	8	9	10
8	4	5	1
9	2	4	10
10	1	1	1

Es werden nicht immer alle möglichen Werte durchlaufen!

Untergruppen

Problem: Potenz durchläuft nicht alle möglichen Werte.

- Lösung nicht eindeutig (Entschlüsselung nicht möglich?)
- Durchprobieren führt schneller zum Ziel

Müssen a geeignet wählen – geht das immer?

Untergruppen

Problem: Potenz durchläuft nicht alle möglichen Werte.

- Lösung nicht eindeutig (Entschlüsselung nicht möglich?)
- Durchprobieren führt schneller zum Ziel

Müssen a geeignet wählen – geht das immer?

- Die Potenzen von a bilden immer eine *Untergruppe*.
- Die Anzahl der Zahlen einer Untergruppe ist immer ein Teiler der ursprünglichen Gruppengröße (Satz von Lagrange).
- Wenn also Gruppengröße Primzahl p ist, kann die Untergruppe nur 1 oder p Elemente enthalten.
- Multiplikative Gruppe hat aber immer $p - 1$ (gerade Anzahl) Elemente.
- Es existiert trotzdem immer geeignete Basis a (*Primitivwurzel*).
- Für bestimmte Primzahlen (*Sophie-Germain-P.*) leicht zu finden.

Grundidee

- Benutzen Körper $\mathbf{Z}_p$
- a ist bekannt, Exponent x ist geheimer Schlüssel, $c = a^x$ wird zum Verschlüsseln benutzt.
- Für Entschlüsseln muss x bekannt sein, Kryptoanalyse muss also diskreten Logarithmus berechnen.

Verfahren:

- Diffie-Hellman-Schlüsseltausch (DHE)
- El-Gamal-Public-Key-Verfahren
- El-Gamal-Signaturverfahren

Modell Diffie-Hellman-Schlüsseltausch

- Große Zahl von verschiedenen Farben vorhanden, können gemischt, aber nicht entmischt werden.
- Alice und Bob wollen die gleiche geheime Mischfarbe produzieren, Eve überwacht aber die Kommunikation.
- Alice würfelt Farbe A aus und teilt diese Bob mit (nicht geheim).
- Alice wählt geheime Farbe x, mischt zu gleichen Teilen mit A und übergibt diese Farbe Ax Bob (Eve kann Proben abzweigen!).
- Bob wählt geheime Farbe y, mischt zu gleichen Teilen mit A und übergibt diese Farbe Ay an Alice.
- Bob kann nun aus Ax und seiner geheimen Farbe y die geheime Farbe Axy mit gleichen Farbanteilen mischen.
- Alice kann aus Ay und geheimer Farbe x ebenfalls die geheime Farbe Axy mischen.
- Alice und Bob haben beide geheime Farbe Axy, die nie ausgetauscht werden musste.
- Eve kann zwar $AAxy$, aber nicht Axy mischen und die geheime Farbe damit nicht herstellen.

Diffie-Hellman-Schlüsseltausch

- Alice wählt eine genügend große Primzahl p und eine Primitivwurzel A und teilt diese Bob mit.
- Alice wählt geheimen Exponent x, berechnet A^x und teilt diesen Wert Bob mit.
- Bob wählt geheimen Exponent y, berechnet A^y und teilt diesen Wert Alice mit.
- Alice berechnet $(A^y)^x = A^{x \cdot y}$ und hat den geheimen Schlüssel.
- Bob berechnet $(A^x)^y = A^{x \cdot y}$ und hat ebenfalls den geheimen Schlüssel.
- Eve kennt zwar A^x und A^y, kann durch Verknüpfen beider Werte nie $A^{x \cdot y}$ berechnen.

Das El-Gamal-Public-Key und -Signaturverfahren benutzen gleiche Grundidee.

Um das Verfahren sicher zu machen, müssen auch x und y gewisse Bedingungen erfüllen.

Andere Verfahren mit diskreten Logarithmus

Warum?

- Berechnung des diskreten Logarithmus noch schwieriger machen.
- Berechnung schneller machen.
- Ersatzverfahren, falls diskreter Logarithmus in Grundform doch schnell berechnet werden kann.

Ideen:

- Andere Gruppenoperation?
- Anderer endlicher Körper?

$\Longrightarrow$ Elliptische Kurven!

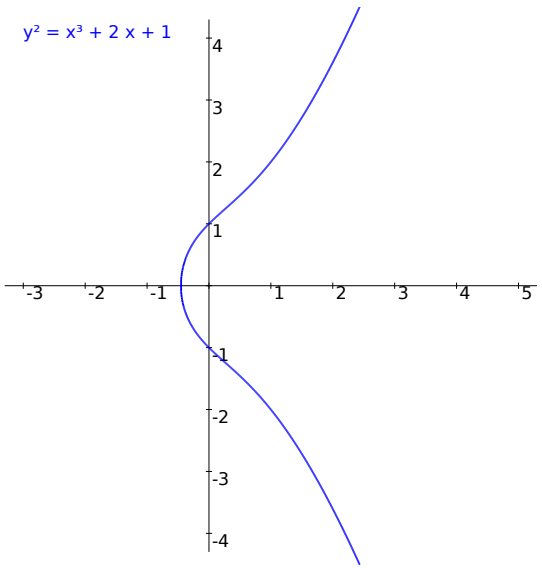
Elliptische Kurven in der Ebene

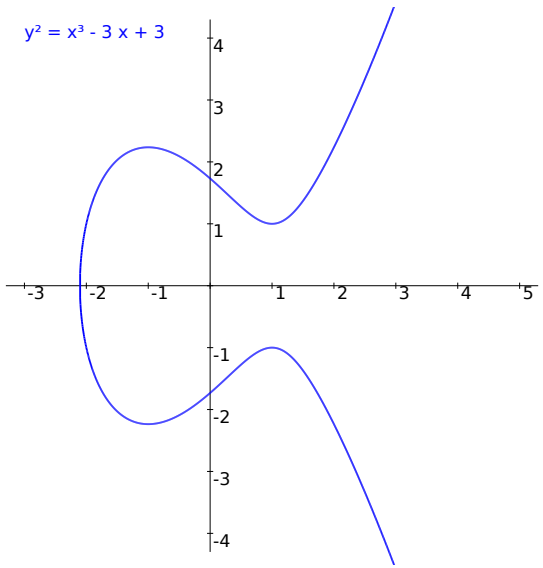
- (x, y) sind Punkte der (kartesischen) Ebene
- Betrachen alle Punkte, deren Koordinaten die Gleichung

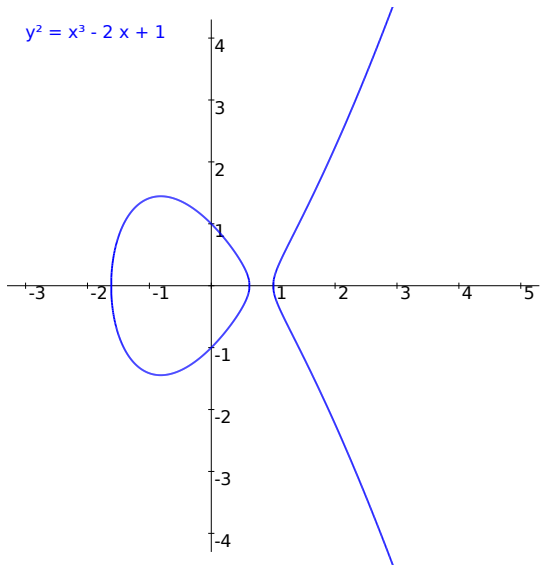
$$y^2 = x^3 + a \cdot x + b$$

  erfüllen. a, b sind dabei reelle Zahlen.
- Nur lösbar, wenn rechte Seite nichtnegativ, dann zwei Lösungen
- Kurve symmetrisch zur x-Achse
- Kurve hat 1 oder 3 Nullstellen

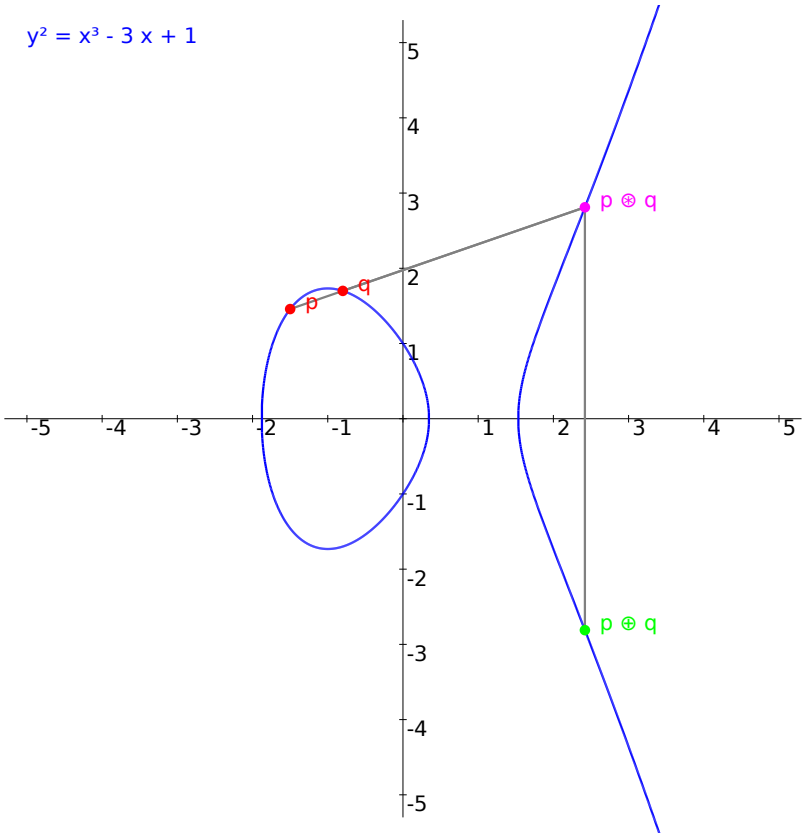
Beispiele elliptischer Kurven





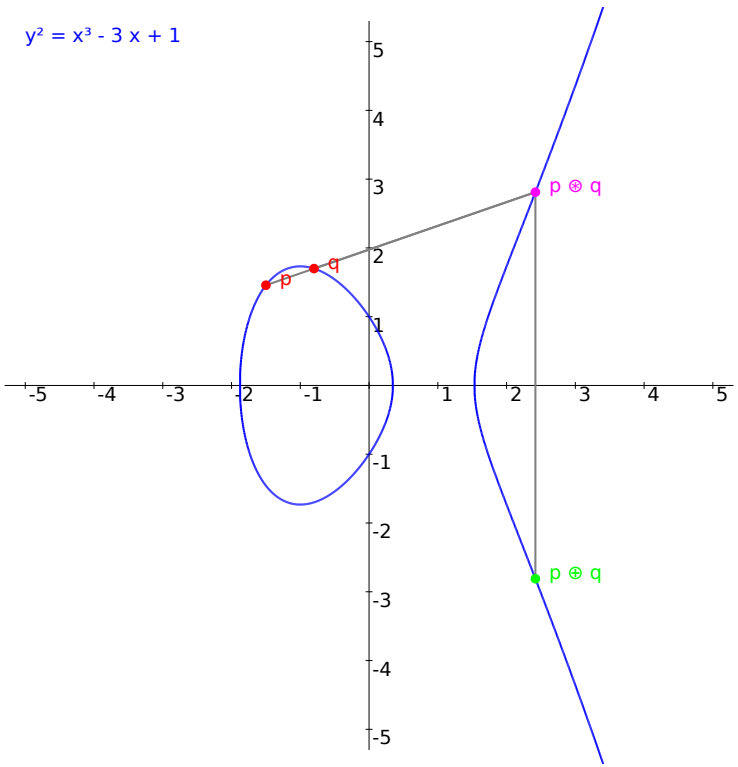


Kurvenoperation $\oplus$



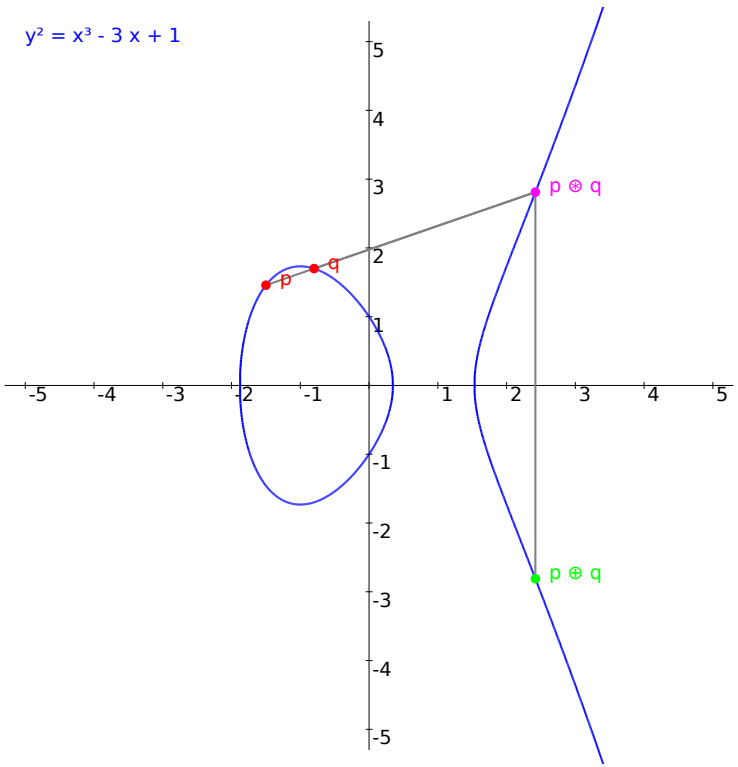
- Verbinden zwei Kurvenpunkte p, q durch Linie
- Falls $p = q$, dann Tangente
- Linie schneidet Kurve in drittem Punkt $p \circledast q$
- Spiegeln diesen Punkt an x-Achse und erhalten $p \oplus q$

Gruppenoperation $\oplus$



- Wo ist die Null?
- Inverses Element?
- Ist $p \oplus q$ immer Kurvenpunkt?
- Assoziativgesetz?
- Kommutativgesez?
  $\Longrightarrow$ Ja, offensichtlich.

Gruppenoperation $\oplus$



- Wo ist die Null?
  $\Longrightarrow$ Fernpunkt in y-Richtung
  (Herleitung über projektive Darstellung)
- Inverses Element?
  $\Longrightarrow$ Spiegelung an x-Achse
- Ist $p \oplus q$ immer Kurvenpunkt?
  $\Longrightarrow$ Ja, lässt sich zeigen.
- Assoziativgesetz?
  $\Longrightarrow$ Ja, längere Rechnung.
- Kommutativgesez?
  $\Longrightarrow$ Ja, offensichtlich.

Bestimmen von $p \oplus q$

Lange Rechnung liefert verblüffend einfache Formel
($p = (x_1, y_1)$, $q = (x_2, y_2)$, $p \oplus q = (x_3, y_3)$):

$$x_3 = \lambda \cdot \lambda - x_1 - x_2$$
$$y_3 = \lambda \cdot (x_1 - x_3) - y_1$$

mit

$$\lambda = \begin{cases} \frac{y_2 - y_1}{x_2 - x_1}, & \text{falls } p \neq q \\ \frac{3 \cdot x_1 \cdot x_1 + a}{2 \cdot y_1}, & \text{falls } p = q \end{cases}$$

- Nur Grundrechenarten (also Körperoperationen) verwendet
- Wenn p und q rational, dann auch $p \oplus q$

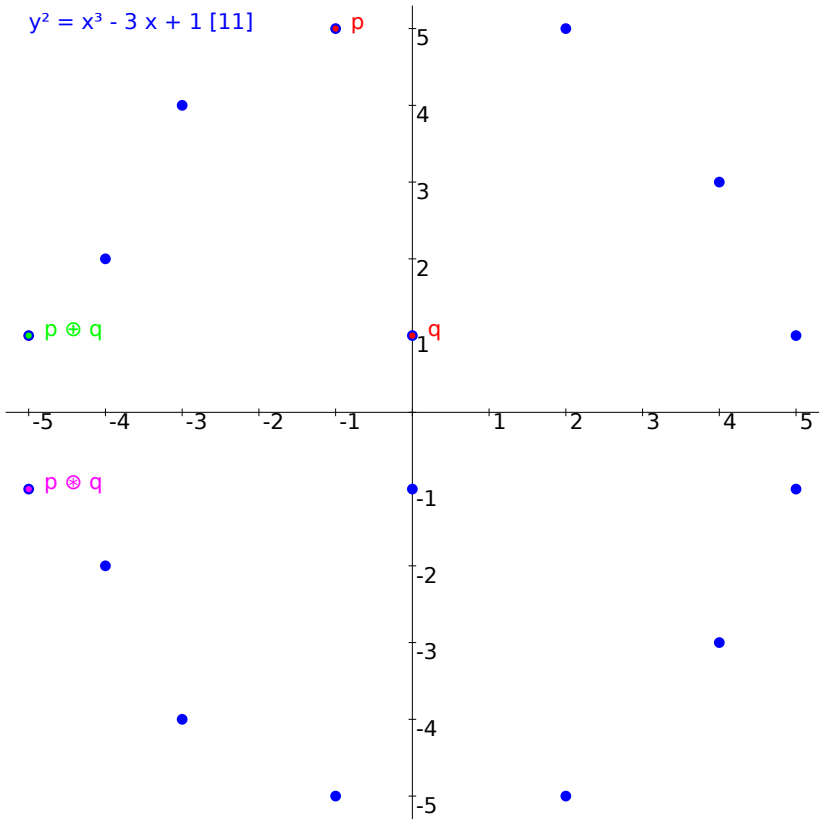
Betrachten wieder die Kurvengleichung

$$y^2 = x^3 + a \cdot x + b$$

- Wir ersetzen Punktkoordinaten und Kurvenparameter durch Zahlen aus Z_p.
- Wir berechnen $p \oplus q$ nach gleicher Formel wie bisher.
- Rechnung möglich, wir erhalten wieder «Kurvenpunkt» (erfüllt $y^2 = x^3 + a \cdot x + b$)
- Geometrische Interpretation nicht mehr möglich.

Elliptische Kurve in Z_{11}

Symmetrisch dargestellt, $-x = 11 - x$



Diskreter Logarithmus auf elliptischer Kurve

Idee (Diffie Hellman):

- Alice und Bob wählen gemeinsame elliptische Kurve.
- Alice wählt *geeigneten* Kurvenpunkt P.
- Alice wählt geheimen Exponenten x,
  berechnet Punkt $Q = P(\oplus)^x$
- Alice teilt Bob P und Q mit.
- Bob berechnet $R = P(\oplus)^y$ mit geheimen Schlüssel y.
- Alice erhält R und berechnet
  geheimen Schlüssel $R(\oplus)^x = P(\oplus)^{x \cdot y}$.
- Bob berechnet aus Q geheimen Schlüssel $Q(\oplus)^y = P(\oplus)^{x \cdot y}$.

Unser geheimer Schüssel ist nun ein *Punkt*, also ein Zahlenpaar.

Punktzahl der elliptischen Kurve

Fragen:

- Gibt es (außer der $\mathcal{O}$) immer Punkte auf der elliptischen Kurve?
- Wenn ja, wieviele?

Punktzahl der elliptischen Kurve

Fragen:

- Gibt es (außer der $\mathcal{O}$) immer Punkte auf der elliptischen Kurve?
  Ja
- Wenn ja, wieviele?
  Satz von Haase:

$$p + 1 - 2 \cdot \sqrt{p} \leq \text{Punktzahl} \leq p + 1 + 2 \cdot \sqrt{p}$$

- Punktzahl ist also ungefähr p.
- Können wir Punktzahl exakt berechnen?
  Am besten wäre eine Primzahl $\Longrightarrow$ Potenzen *jedes* Kurvenpunktes durchlaufen *alle* Kurvenpunkte

Shoof-Algorithmus

Exakte Bestimmung der Punktzahl der Kurve.

Algorithmusidee:

- Bestimmen nicht die Zahl der Kurvenpunkte, sondern Differenz zu p: $\approx \pm 2 \cdot \sqrt{p}$
- Zahl wird nicht direkt bestimmt, sondern der Divisionsrest dieser Zahl modulo 2, 3, 5, ... (Primzahlen)
- Benutze soviele Primzahlen, bis deren Produkt größer als $4 \cdot \sqrt{p}$ ist.
- Nach chinesischem Restsatz wird daraus gesuchte Zahl berechnet.

Genauer Algorithmus schwierig, aber «schnell» berechenbar: $(\log p)^9$ nach [4] bzw. $(\log p)^8$ nach [5].

Schnelleres Verfahren: SEA: $(\log p)^6$ [5].

Bestimmung einer geeigneten Kurve

Prinzipielles Vorgehen:

1. Auswürfeln einer genügend großen Primzahl p.
2. Auswürfeln der Kurvenparamater a und b (gewisse Einschränkungen für sichere Kurven).
3. Bestimmen der Anzahl der Kurvenpunkte mit Shoof-Algorithmus.
4. Falls Anzahl keine Primzahl, zurück zu (2).
5. Berechnen eines beliebigen Kurvenpunktes P: Wurzelberechnung in endlichem Körper schwierig. $\Longrightarrow$ ebenfalls mit Shoof-Algorithmus möglich.

Alternative: Fertige Kurven des NIST

NIST: National Institute of Standards and Technologie

FIPS PUB 186-3

FEDERAL INFORMATION PROCESSING STANDARDS PUBLICATION

Digital Signature Standard (DSS)

CATEGORY: COMPUTER SECURITY **SUBCATEGORY: CRYPTOGRAPHY**

Information Technology Laboratory
National Institute of Standards and Technology
Gaithersburg, MD 20899-8900

Issued June, 2009

U.S. Department of Commerce
Gary Locke, Secretary
National Institute of Standards and Technology
Patrick Gallagher, Deputy Director

Anhang D: Empfohlene Kurven ($a = -3$)

- The coefficient b (satisfying $b^2\, c \equiv -27 \pmod{p}$)
- The base point x coordinate G_x
- The base point y coordinate G_y

The integers p and n are given in decimal form; bit strings and field elements are given in hexadecimal.

D.1.2.1 Curve P-192

$p =$ 6277101735386680763835789423207666416083908700390324961279

$n =$ 6277101735386680763835789423176059013767194773182842284081

$SEED =$ 3045ae6f c8422f64 ed579528 d38120ea e12196d5

$c =$ 3099d2bb bfcb2538 542dcd5f b078b6ef 5f3d6fe2 c745de65

$b =$ 64210519 e59c80e7 0fa7e9ab 72243049 feb8deec c146b9b1

$G_x =$ 188da80e b03090f6 7cbf20eb 43a18800 f4ff0afd 82ff1012

$G_y =$ 07192b95 ffc8da78 631011ed 6b24cdd5 73f977a1 1e794811

D.1.2.2 Curve P-224

$p =$ 26959946667150639794667015087019630673557916260026308143510066298881

$n =$ 26959946667150639794667015087019625940457807714424391721682722368061

$SEED =$ bd713447 99d5c7fc dc45b59f a3b9ab8f 6a948bc5

$c =$ 5b056c7e 11dd68f4 0469ee7f 3c7a7d74 f7d12111 6506d031 218291fb

$b =$ b4050a85 0c04b3ab f5413256 5044b0b7 d7bfd8ba 270b3943 2355ffb4

$G_x =$ b70e0cbd 6bb4bf7f 321390b9 4a03c1d3 56c21122 343280d6 115c1d21

$G_y =$ bd376388 b5f723fb 4c22dfe6 cd4375a0 5a074764 44d58199 85007e34

88

Alternative Kurven

- Kein Vertrauen zu amerikanischen Behörden
- Berechnungen zu komplex $\Longrightarrow$ fehlerhafte Implementierungen
- Suche nach einfacheren Kurven
- Daniel J. Bernstein: Curve25519 (2006)
  Implementierung verfügbar, häufig eingesetzt:

$$p = 2^{255} - 19$$
$$y^2 = x^3 + 486662 \cdot x^2 + x$$

- Anderer Gleichungstyp!

Empfohlene Schlüssellängen

Länge der Primzahl p in Bit:

Symmetrisch	RSA / Diffie-Hellman	Elliptische Kurven
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	521

Quelle: NSA[9]

Angriffe

3 Gruppen:

- Allgemeine Verfahren für diskreten Logarithmus:
  - Babystep-Giantstep-Algorithmus
  - Pollard-ρ-Verfahren
  - 2014 neue Verfahren (nur Spezialfälle oder allgemein?)

  Laufzeit: $O(\sqrt{\text{Gruppengröße}})$
- Spezielle Verfahren für elliptische Kurven:
  - MOV-Algorithmus
  - SSSA-Algorithmus
  - Weitere Verfahren?

  Funktionieren nur für spezielle Kurven

  $\Longrightarrow$ geeignete Kurve wählen
- Quantencomputer
  - Aufwand proportional zu Schlüssellänge
  - Schlüssel deutlich kürzer als bei RSA

Galoisfelder

- Zusammenfassende Bezeichnung *aller* endlichen Körper.
- Es gibt erstaunlich wenig endliche Körper!
- Elementzahl ist stets $q = p^k$, wobei p Primzahl und k beliebige natürliche Zahl ist.

Struktur von Galoisfeldern

Betrachten Polynome $(k-1)$-ten Grades mit Koeffizienten aus $\mathbf{Z}_p$:

$$p_{k-1}(X) = a_{k-1}X^{k-1} + a_{k-2}X^{k-2} + \cdots + a_1X + a_0$$

- Es gibt davon p^k viele, die durch ihr Koeffiziententupel $(a_0, a_1, \ldots, a_{k-1})$ beschrieben sind
  $\Longrightarrow$ Das sind unsere «Zahlen»!
- X ist nur Symbol, hier wird nie ein Wert eingesetzt.
- Zusätzlich *irreduzibles Polynom* k-ten Grades.

$$I_k = X^k + r_{k-1}X^{k-1} + \cdots + r_1X + r_0$$

  Kann nicht in Faktoren (bzgl. $\mathbf{Z}_p$) zerlegt werden!
- I_k ist das Aquivalent von p der Primzahlkörper
- Es gibt «genügend viele» irreduzible Polynome für jedes p, k.

Rechnen im Galoisfeld (1)

Beispiel: $GF(2^3)$

- Irreduzibles Polynom z. B. $I_3 = X^3 + X + 1$
- 8 Elemente:
  $0, 1, X, X+1, X^2, X^2+1, X^2+X, X^2+X+1$
- Addition, Subtraktion:

$$\begin{aligned}(X^2+1)+(X+1) &= X^2+X+2 = X^2+X+0\\ &= X^2+X\\ (X^2+1)-(X^2+X+1) &= -X\\ &= X\\ (X^2+X)+(X^2+X) &= 2\cdot X^2+2\cdot X = 0\end{aligned}$$

- Beobachtung: Jedes Element ist zu sich selbst invers.

Rechnen im Galoisfeld (2)

- Multiplikation:

$$(X+1)\cdot(X+1) = X^2 + 2\cdot X + 1$$
$$= X^2 + 1$$

$\Longrightarrow X^2 + 1$ ist in $GF(2^k)$ nie irreduzibel!

- Bei Überlauf: Vielfache von I_k abziehen:

$$(X^2+1)\cdot(X^2+X+1)$$
$$= X^4 + X^3 + 2\cdot X^2 + X + 1$$
$$= X^4 + X^3 + X + 1 \qquad | -X\cdot(X^3+X+1) = X^4 + X^2 + X$$
$$= X^3 - X^2 + 1$$
$$= X^3 + X^2 + 1 \qquad | -(X^3+X+1)$$
$$= X^2 - X = X^2 + X$$

- Inverses Element: Wieder mit Euklidschem Algorithmus.
- Damit Division durch Multiplikation mit inversem Polynom.

Besonders interessant: $p = 2$

$\Longrightarrow$ Schnelle Hardwarelösungen (Chipkarten)

Elliptische Kurven in Galoisfeldern

- Für $GF(2^k)$ ist Kurve

$$y^2 = x^3 + a \cdot x + b$$

  unsicher (supersingulär) (SSSA-Algorithmus)
- Müssen auf kompliziertere Kurve

$$y^2 + x \cdot y = x^3 + a \cdot x + b$$

  ausweichen.
- x, y, a, b sind Elemente aus $GF(p^k)$, also Polynome.
- Gleichungen zur Berechnung des inversen Punktes und dritten Kurvenpunktes etwas komplizierter.
- Alle weiteren Rechnungen verlaufen analog

Quellen I

[1] Buchmann: Einführung in die Kryptographie. Springer 2004.

[2] Werner: Elliptische Kurven in der Kryptographie. Springer 2002.

[3] Kurzweil: Endliche Körper. Springer 2007.

[4] Schoof: Elliptic Curves Over Finite Fields and the Computationof Square Roots mod p. Mathematics of Computation Vol. 44, No. 170.

[5] Mirbach: Elliptische Kurven; die Bestimmung ihrer Punktezahl und Anwendung in der Kryptographie. Monsenstein und Vannerdat 2003.

[6] Bernstein: A state-of-the-art Diffie-Hellman function. `http://cr.yp.to/ecdh.html`

Quellen II

[7] Bernstein: Curve25519: new Digitale-Hellman speed records. http://cr.yp.to/ecdh/curve25519-20060209.pdf

[8] NIST: Digital Signature Standard (DSS). FIPS PUB 186-3.

[9] The Case for Elliptic Curve Cryptography. (NSA) https://www.nsa.gov/business/programs/elliptic_curve.shtml